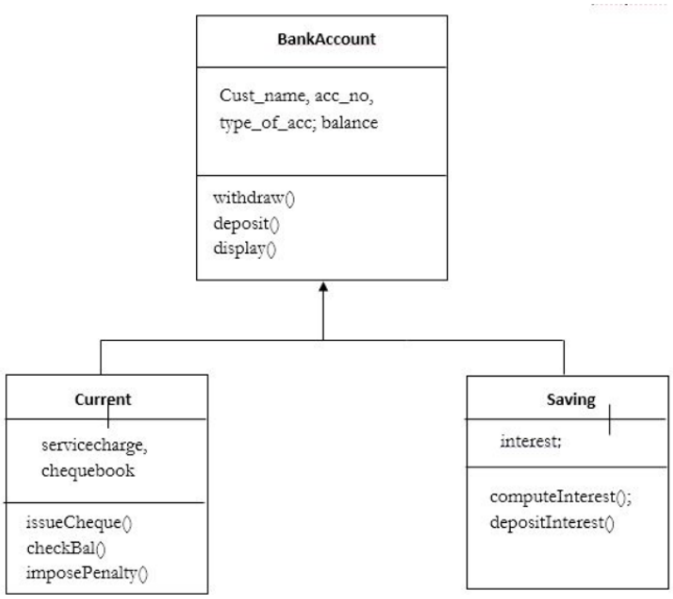
Assignment 1- CSE 103

1> Class Diagram –



Program:

```
#include<iostream.h>
#include<stdio.h>
#include<math.h>
#include<conio.h>
#include<string.h>

class account{

protected:
    char cname[20];
    int accno;
    char type;
```

```
    int bal;
public:
    account()
    {
      strcpy(cname," ");
      accno=0;
      type=' ';
      bal=0;
    }
    void input(){
      cout<<"Enter cname";cin>>cname;
      cout<<"Enter accno";cin>>accno;
      fflush(stdin);
      cout<<"Enter type"; cin>>type;
      fflush(stdin);
      cout<<"Enter bal";cin>>bal;
    }
    void display(){
        cout<<"\n Customer Name "<<cname;
        cout<<"\n Account Number "<<accno;
        cout<<"\n Type "<<type;
        cout<<"\n Balance "<<bal;
    }
    void deposit(){
        int amt;
        cout<<"\n Enter the amount to deposit";
        cin>>amt;
        bal=bal+amt;
    }
};
class savacct:public account{
```

```
    int inter;
    public:

    int comp_int(){
      int time1,rate1;
      rate1=10;
      cout<<"\n Enter time";cin>>time1;
      inter=bal*pow(1+rate1/100.0,time1)-bal;
      return inter;
    }

    void update_bal(){
      bal=bal+comp_int();
    }

    void withdrawal(){
      int amt;
      cout<<"\n Enter amount to withdrawn";
      cin>>amt;
      if(bal>=amt){
          bal=bal-amt;
      }
      else{
          cout<<"\n The amount cannot be withdrawn";
      }
    }
  };

class curacct:public account{
     int chq_bk;
     int penal;
```

```
    public:

    int min_bal(){
      int ret1=1;
      if(bal<=500){
        penal=50;
        bal=bal-penal;
        ret1=0;
      }
      else{
        cout<<"\n No penality imposed";
      }
      return ret1;
    }
    void withdrawal(){
      int amt;
      cout<<"\n Enter the amount to withdrawn";
      cin>>amt;
      int k=min_bal();
      if(k==1){
        if(bal>=amt)
        bal=bal-amt;
      }
      else{
         cout<<"\n The amount cannot be withdrawn";
      }
    }
  };

void main(){
    curacct c1;
```

```
    savacct s1;
    c1.input();
    c1.display();
    c1.deposit();
    c1.display();
    c1.withdrawal();
    c1.display();
    s1.input();
    s1.display();
    s1.deposit();
    s1.display();
    s1.withdrawal();
    s1.display();
}
```

2> UML Class Diagram –